

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais
de Eletromiografia de Superfície para
Classificação de Movimentos Utilizando Redes
Neurais Artificiais**

Porto Alegre

2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Eng. Alexandre Balbinot

Porto Alegre

2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Trabalho aprovado. Porto Alegre, XX de XXXXXX de 2015:

Prof. Dr. Eng. Alexandre Balbinot
Orientador

Professor
Convidado 1

Professor
Convidado 2

Porto Alegre
2015

*A Gilberto, meu pai, torre de razão.
Acaso lo que digo no es verdadero, ojalá sea profético.*

Agradecimentos

Aos demais colaboradores e pesquisadores do laboratório de Instrumentação Eletro-Eletrônica, em especial Vinicius Cene e Fernanda Trevisol, que desenvolveram a coleta da base de dados utilizada e prestaram auxílio de forma geral.

*Take nothing on its looks;
take everything on evidence.
There's no better rule.*

Charles Dickens, Great Expectations

Resumo

A segmentação de sinais de eletromiografia (EMG) é parte essencial de pré-processamento em aplicações de reconhecimento de movimentos e controle de próteses, separando trechos de interesse do sinal correspondentes a esforços musculares e descartando trechos de sinal com baixa atividade muscular. Neste estudo, quatro métodos para segmentação automática de sinais de EMG foram implementados em MATLAB. Os métodos foram aplicados aos sinais de EMG de superfície a base de dados do projeto Ninapro e aos sinais da base de dados adquiridos pelo Laboratório de Instrumentação Eletro-Eletrônica da UFRGS. Uma rede neural artificial foi utilizada para classificar os movimentos realizados correspondentes aos sinais segmentados com os quatro métodos. TODO: RESULTADOS.

Palavras-chave: Eletromiografia. Segmentação. Base de dados Ninapro. Redes Neurais Artificiais.

Abstract

TRADUZIR RESUMO

Keywords: Eletromiography. Segmentation. MATLAB. Ninapro database. Artificial Neural Networks.

Lista de Figuras

Figura 1 – Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$	15
Figura 2 – MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.	16
Figura 3 – Fluxograma representativo do MTD1.	18
Figura 4 – Fluxograma representativo do MTD2.	19
Figura 5 – Fluxograma representativo do MTD3.	22
Figura 6 – Fluxograma representativo do MTD4.	23
Figura 7 – Fluxograma geral para os métodos de segmentação MTD1 - MTD4. . .	24
Figura 8 – Retificação completa de trecho de sinal de eletromiografia.	25
Figura 9 – Normalização de canais de eletromiografia de acordo com seu valor máximo.	26
Figura 10 – <i>Clustering</i> por <i>k-means</i> dos centros de segmentos obtidos pelos métodos MTD1 e MTD2.	27
Figura 11 – <i>Clustering</i> por <i>k-means</i> de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.	28
Figura 12 – <i>Clustering</i> por <i>k-means</i> de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.	29

Lista de Tabelas

Tabela 1	–	Parâmetros utilizados para definir MTD1.	17
Tabela 2	–	Parâmetros utilizados para definir MTD2.	19
Tabela 3	–	Parâmetros utilizados para definir MTD3.	20
Tabela 4	–	Parâmetros utilizados para definir MTD4.	21
Tabela 5	–	Parâmetros ajustáveis para os métodos de segmentação.	26

Lista de abreviaturas e siglas

EMG	Eletromiografia
MU	<i>Motor Unit</i>
MUAP	<i>Motor Unit Action Potencial</i>
MUAPT	<i>Motor Unit Action Potencial Trains</i>
MTD#	<i>Método número #</i>
BEP	<i>Beginning Extraction Point</i>
EEP	<i>Ending Extraction Point</i>

Sumário

1	INTRODUÇÃO	13
2	REVISÃO BIBLIOGRÁFICA	15
2.1	Sinais de Eletromiografia	15
2.2	Métodos de Segmentação	16
2.2.1	Método 1 - método iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD1)	17
2.2.2	Método 2 - método não iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD2)	18
2.2.3	Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)	20
2.2.4	Método 4 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando <i>thresholding</i> (MTD4)	20
2.3	Bases de Dados Utilizadas	21
2.3.1	Base de Dados Ninapro	21
2.3.1.1	Base de Dados Ninapro	21
2.3.2	Base de Dados do IEE	21
3	METODOLOGIA EXPERIMENTAL	24
3.1	Implementação de Métodos de Segmentação	24
3.1.1	Preprocessamento	24
3.1.1.1	Retificação Completa	24
3.1.1.2	Normalização	25
3.1.2	Implementação dos Métodos de Segmentação	25
3.1.2.1	Parâmetros Utilizados	25
3.1.2.2	Identificação de segmentos utilizando <i>k-means</i>	27
3.2	Rede Neural Artificial	27
3.2.1	Características Utilizadas como Preditores	28
3.2.2	Movimentos Utilizados como Alvo	29
4	RESULTADOS E DISCUSSÕES	30
5	CONCLUSÕES	31
6	PROPOSTAS DE FUTUROS TRABALHOS	32

REFERÊNCIAS	33
 APÊNDICES	 35
APÊNDICE A – FUNÇÃO EM MATLAB PARA MTD1	36
APÊNDICE B – FUNÇÃO EM MATLAB PARA MTD2	39
APÊNDICE C – CÓDIGO UTILIZADO PARA DETERMINAÇÃO DE r_{target} DO MTD1	42
APÊNDICE D – FUNÇÃO EM MATLAB PARA OBTENÇÃO DOS MOVIMENTOS RELACIONADOS AOS SEGMENTOS	43

1 Introdução

Sinais de EMG apresentam crescentes aplicações no controle de próteses mioelétricas. Por exemplo, (HARGROVE et al., 2013) mostra o controle de uma prótese de perna de um amputado acima do joelho direito, enquanto (CHU et al., 2007) apresentou bons resultados de reconhecimento de padrões de EMG para desenvolvimento de uma prótese multifuncional de mão. Utilizando redes neurais artificiais, (PATTICHIS; SCHIZAS; MIDDLETON, 1995) mostra realização de diagnósticos clínicos de desordens neuromusculares em sinais de EMG invasivo (adquirido por agulhas).

As principais estratégias para caracterização de sinais de EMG e potenciais de ação das unidades motoras baseiam-se no uso de um método classificador. Métodos de classificação utilizados incluem - entre inúmeros outros - redes neurais artificiais (HUDGINS; PARKER; SCOTT, 1993), classificador Bayesiano (ENGLEHART; HUDGINS, 2003) e lógica *fuzzy* (CHAN et al., 2000). Tais sistemas de classificação necessitam, como parte do pré-processamento, segmentar os sinais de EMG adquiridos, para então realizar extração de características dos segmentos como amplitude, número de cruzamentos por zero, coeficientes de autoregressão, transformadas de Fourier e, mais recentemente, transformadas Wavelet (CHU et al., 2007).

Neste trabalho, propõe-se e implementa-se em MATLAB quatro diferentes métodos de segmentação automática para sinais de EMG de superfície. Os métodos serão aplicados em sinais da base de dados número 2 do projeto Ninapro (ATZORI et al., 2014) - que contém sinais de EMG de superfície em 12 canais para 49 diferentes movimentos de mão e punho - e em sinais da base de dados similarmente adquiridos pelo Laboratório de Instrumentação Eletro-Eletrônica.

Os primeiros dois métodos (que serão identificados neste estudo pelos mnemônicos MTD1 e MTD2) tratam da detecção de picos do sinal utilizando *thresholding* e produzem segmentos de comprimento constante centrados nestes picos. O terceiro (MTD3) e quarto (MTD4) métodos utilizam de janela deslizante para identificação de pontos iniciais e finais dos segmentos, produzindo segmentos de comprimento variável. Para as implementações dos métodos neste estudo, assume-se o conhecimento *a priori* do número de segmentos de interesse contidos no sinal.

O primeiro método (MTD1) é baseado no método de segmentação utilizado em (CHAUVET et al., 2001). Trata-se de método iterativo, identificando os picos do sinal a partir de *threshold* de amplitude, segmentando o sinal em janelas de comprimento constante centradas nos picos. O valor de *threshold* para a primeira iteração corresponde ao máximo

absoluto do sinal. A cada nova iteração em que não se atinge um número desejado de segmentos, o novo *threshold* é calculado como fração do *threshold* da iteração anterior.

O segundo método (MTD2) é baseado no método de segmentação utilizado em (KATSIS et al., 2006). De forma similar ao MTD1, também utiliza *threshold* para detecção de picos do sinal e segmentação com janelas de comprimento constante em torno dos picos. Diferentemente de MTD1, MTD2 não é iterativo, utilizando o valor máximo e o comprimento do sinal para cálculo do valor de *threshold*.

O terceiro método (MTD3) é baseado no método de segmentação utilizado em (GUT; MOSCHYTZ, 2000). Uma janela deslizante percorre o sinal e identifica inícios de segmentos quando a declividade média no interior da janela excede determinado valor limite. Os finais dos segmentos são identificados quando a variação total do sinal no interior da janela é inferior a um segundo valor de *threshold*.

O quarto método (MTD4) é baseado no método de segmentação utilizado em (PATTICHIS; SCHIZAS; MIDDLETON, 1995). Os pontos de início do segmento são tais que, em uma janela à esquerda do ponto, o sinal mantém-se abaixo de determinado *threshold*. Os pontos de final de segmento, de forma similar, são tais que, em uma janela à direita do ponto, o sinal mantém-se abaixo do *threshold*.

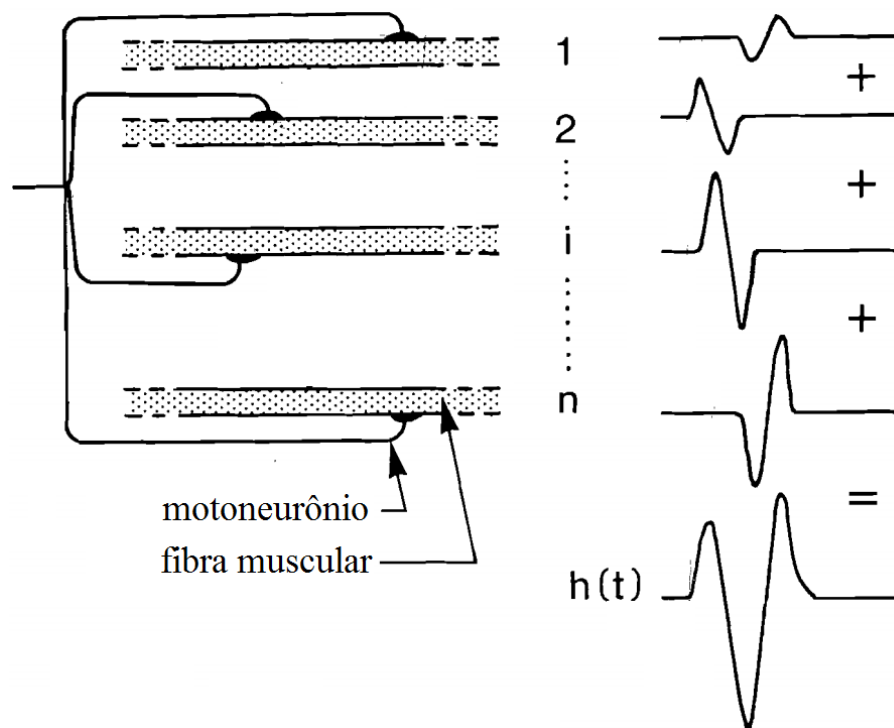
Utilizando valores de RMS, variância e frequência mediana dos sinais segmentados, uma rede neural artificial será treinada para classificar entre os 17 diferentes movimentos de mão e punho constantes do exercício 1 da database número 2 do projeto Ninapro (GIJSBERTS et al., 2014). Os objetivos finais deste trabalho são a implementação dos métodos de segmentação propostos (para possível uso em trabalhos futuros do laboratório de Instrumentação Eletro-Eletrônica) e fornecer avaliação comparativa entre métodos a respeito da taxa de acerto de classificação obtida com uso de RNA.

2 Revisão Bibliográfica

2.1 Sinais de Eletromiografia

Sinais de EMG podem ser adquiridos por eletrodos posicionados na superfície da pele (eletrodo não invasivo) ou por agulhas introduzidas no tecido muscular (eletrodo invasivo). Os sinais são formados por potenciais de ação de fibras musculares organizadas em unidades funcionais chamadas de “unidades motoras” (MU - *Motor Unit*) (LUCA et al., 2006). Uma unidade motora é composta por um neurônio motor e as fibras musculares que ele inerva, e é a entidade fundamental que controla a ativação de músculos estriados (BUCHTHAL; SCHMALBRUCH, 1980). A soma algébrica dos potenciais de ação de todas as fibras de uma unidade motora é chamada de “potencial de ação da unidade motora”, ou em inglês, MUAP (*Motor Unit Action Potential*) (ALMEIDA; BARRETO, 1997). A Figura 1 apresenta a composição de uma MUAP a partir da soma dos potenciais das fibras de uma unidade motora.

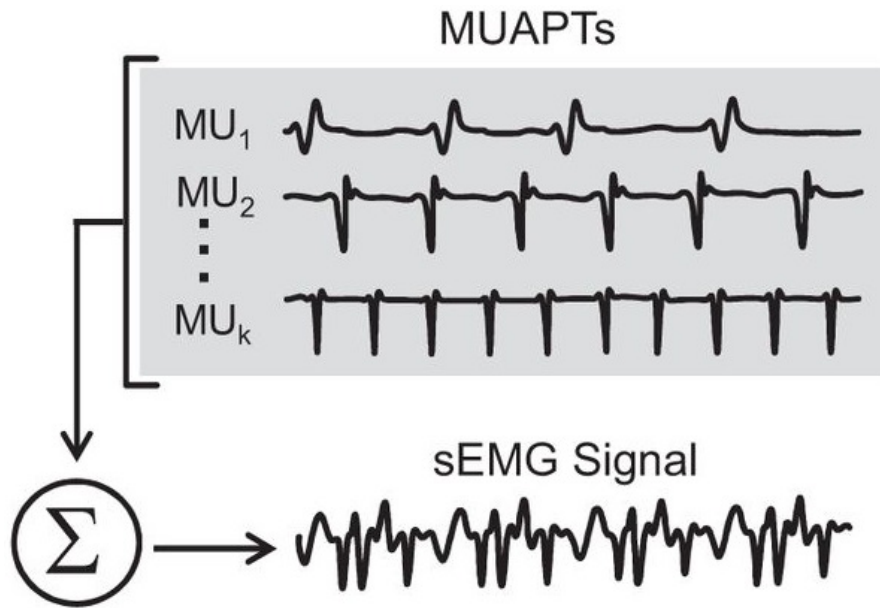
Figura 1: Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$.



Fonte: adaptado de (BASMAJIAN; LUCA, 1985).

Dependendo do método utilizado para aquisição de EMG, é comum a captura da contribuição de mais de uma unidade motora no mesmo canal. A influência de uma unidade motora na amplitude sinal adquirido depende principalmente da distância das fibras musculares ao ponto de aquisição (GERDLE et al., 1999). Sinais de EMG de longa duração são constituídos por sequências temporais de MUAPs, também conhecidas como MUAPTs (*MUAP Trains*). A Figura 2 exemplifica MUAPTs de diferentes MUs que somam-se para formar um sinal de EMG de longa duração.

Figura 2: MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.



Fonte: adaptado de (KLINE; LUCA, 2014)

2.2 Métodos de Segmentação

Esta seção descreve os métodos de segmentação que foram desenvolvidos neste trabalho, citando trabalhos da área que foram utilizados como base teórica para os métodos. Nota-se que nomes utilizados para variáveis e constantes (por exemplo, sinal a ser segmentado ' x ', *threshold* ' T ', etc.) foram determinados pelo autor deste trabalho, não necessariamente sendo estes utilizados nos métodos citados.

Para as definições dos métodos 3 e 4 (MTD3 e MTD4) são utilizados os termos BEP (*beginning extraction point*, ponto inicial de um segmento) e EEP (*ending extraction point*, ponto final de um segmento), também utilizados em (PATTICHIS; SCHIZAS; MIDDLETON, 1995).

2.2.1 Método 1 - método iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD1)

Este método iterativo é adaptado do método de segmentação utilizado em (CHAUVET et al., 2001). As definições da Tabela 1 serão utilizados para descrever este método.

Tabela 1: Parâmetros utilizados para definir MTD1.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T_k	Valor de <i>threshold</i> para a iteração k
T_{lim}	Valor de limite inferior para o <i>threshold</i>
q	Fração de T_{k-1} para determinação de T_k
N_k	Número total de candidatos para centros de segmentos identificados na iteração k
r_k	Razão entre número de candidatos identificados na iteração k e o comprimento total do sinal
r_{target}	Razão mínima esperada para r_k , utilizada para determinar o final do método

Inicialmente, determina-se o valor de *threshold* T_0 equivalente ao máximo absoluto do sinal a ser segmentado x (Equação (2.1)). O valor T_k é atualizado em cada iteração k como sendo uma fração q de T_{k-1} (Equação (2.2)). No trabalho de (CHAUVET et al., 2001), este valor q foi empiricamente determinado em 90%.

$$T_0 = \max(x) \quad (2.1)$$

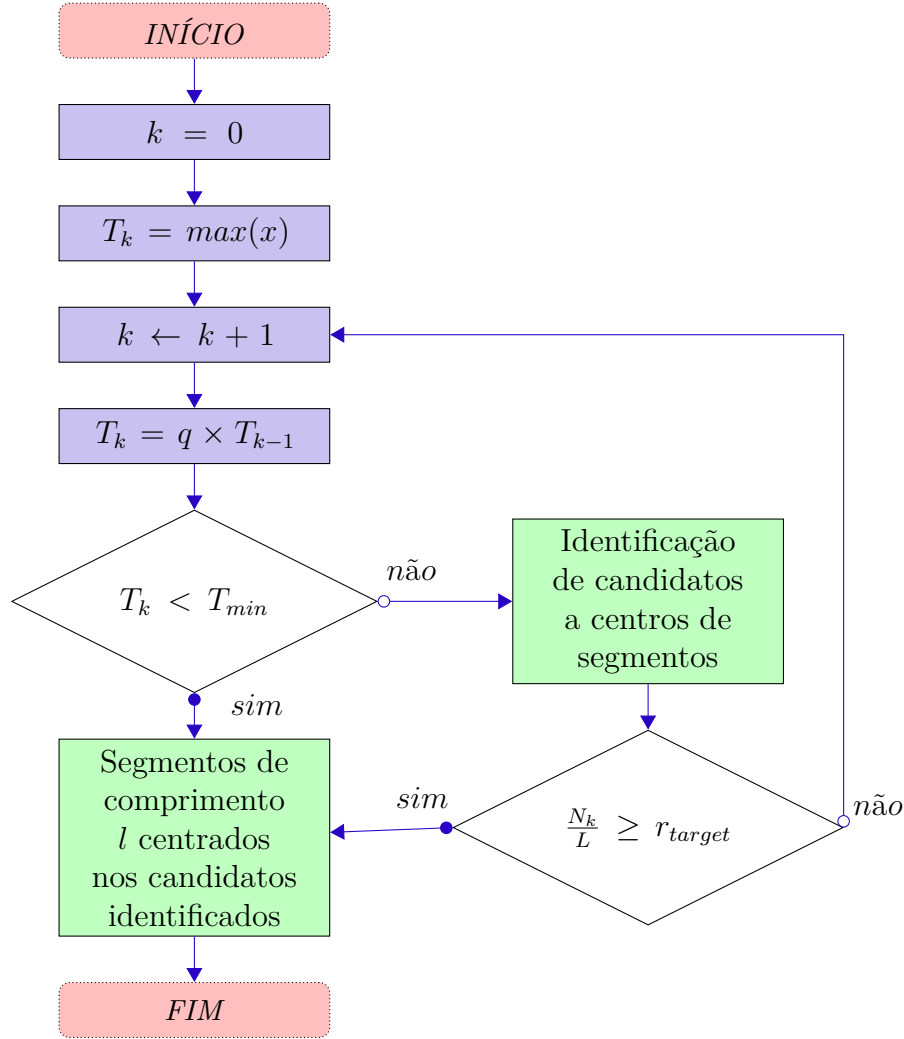
$$T_k = q \times T_{k-1} \quad (2.2)$$

Pontos do sinal acima do valor de T_k são possíveis candidatos para centros de segmentos. Caso exista mais de um possível candidato em uma vizinhança bilateral de l amostras do sinal, apenas o ponto de maior amplitude nesta vizinhança é considerado. Para determinar o final do método, avalia-se a razão r_k entre a quantidade identificada de candidatos N_k e o comprimento total do sinal L (equação 2.3). Caso r_k seja menor que um valor predeterminado r_{target} , calcula-se T_{k+1} para realização da próxima iteração (equação 2.2). Caso r_k seja maior ou igual ao valor predeterminado r_{target} , encerra-se o método e os segmentos são tomados como janelas de sinal de comprimento l , centradas nos candidatos identificados na última iteração.

$$r_k = \frac{N_k}{L} \quad (2.3)$$

Adicionalmente, o estabelecimento de um valor limite mínimo para *threshold* T_{lim} garante que o método não entre em laço infinito e evita detecção de segmentos em trechos de baixa atividade muscular. Caso o valor de *threshold* T_k para a iteração atual seja inferior a T_{lim} , encerra-se o processo iterativo. O método de segmentação MTD1 é representado pelo fluxograma da Figura 3.

Figura 3: Fluxograma representativo do MTD1.



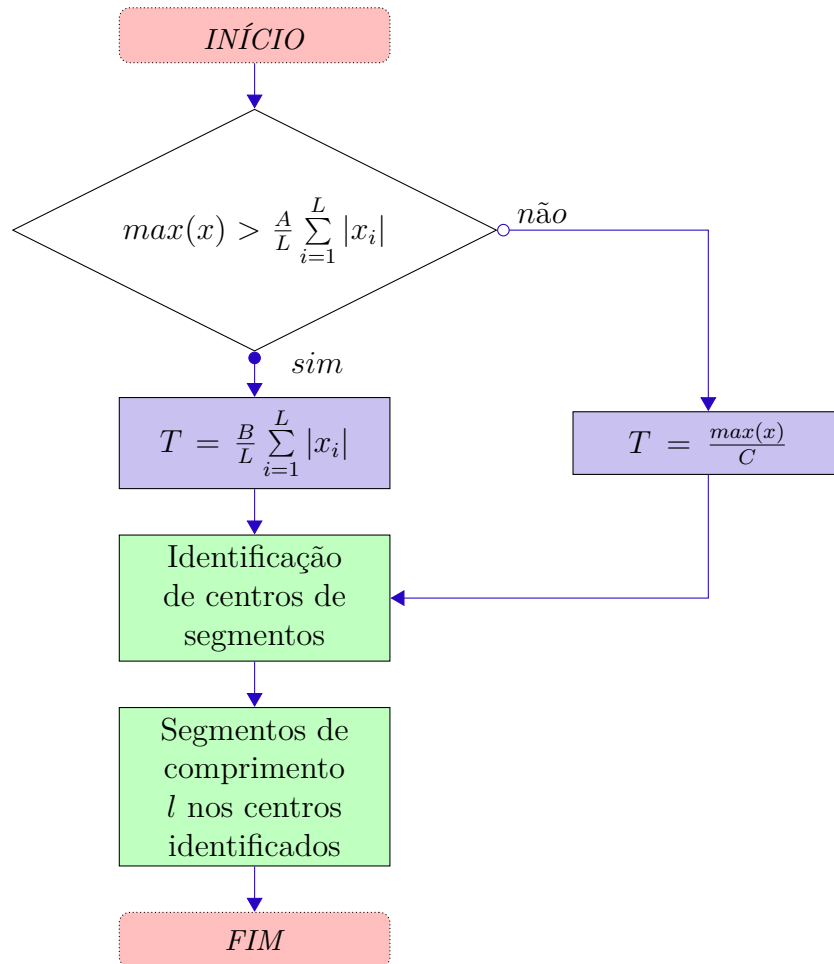
2.2.2 Método 2 - método não iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD2)

Este é o método de segmentação utilizado por (KATSIS et al., 2006), que será descrito pelas definições da Tabela 2. Primeiramente, seleciona-se entre dois métodos de cálculo de *threshold* T : ou utiliza-se T como múltiplo da média aritmética do sinal x ; ou T é uma fração do valor máximo do sinal x . O trabalho de (KATSIS et al., 2006) utiliza a relação do fluxograma da Figura 4 para o cálculo de *threshold* T .

Tabela 2: Parâmetros utilizados para definir MTD2.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T	Valor de <i>threshold</i>
A	Coefficiente utilizado para decisão de método de cálculo de T
B	Múltiplo da média aritmética do sinal x para obtenção de T
C	Fração do valor máximo do sinal x para cálculo de T

Figura 4: Fluxograma representativo do MTD2.



De forma similar ao MTD1, os pontos do sinal que tiverem valor acima de T são considerados possíveis candidatos para centros de segmentos. Para os possíveis candidatos que estiverem afastados de uma distância inferior a l , apenas o candidato de maior amplitude é considerado. Em (KATSIS et al., 2006) utilizou-se coeficientes A , B e C respectivamente de 30, 5 e 5, com comprimento l de 121 amostras.

2.2.3 Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)

Este é o método de segmentação utilizado em (GUT; MOSCHYTZ, 2000). As definições da Tabela 3 serão utilizados para descrever este método.

Tabela 3: Parâmetros utilizados para definir MTD3.

Nome	Descrição
x	Sinal a ser segmentado
W	Comprimento da janela deslizante utilizada pelo método
w_0	Número da amostra mais a esquerda da janela. Determina a posição instantânea da janela
β	Declividade média do sinal x contido na janela deslizante
B	Valor limite para declividade média que determina um BEP
γ	Variação total do sinal x contido na janela deslizante
C	Valor limite para variação total que determina um EEP

Uma janela deslizante de comprimento W percorre o sinal da esquerda para a direita. Caso a declividade média β do trecho de sinal contido pela janela, calculado pela Equação (2.4), exceda um limite B , o ponto mais à esquerda da janela w_0 determina a BEP de um segmento.

$$\beta = \frac{1}{W} \sum_{i=w_0+1}^{w_0+W} (x_i - x_{i-1}) \quad (2.4)$$

O EEP do correspondente segmento é então obtido como o ponto mais à direita da janela ($w_0 + W - 1$) quando a variação total γ , dado pela Equação (2.5), do trecho de sinal contido pela janela for menor que um limite C .

$$\gamma = \sum_{i=w_0+1}^{w_0+W} (x_i - x_{i-1}) \quad (2.5)$$

O MTD3 pode ser representado pelo fluxograma da Figura 5.

2.2.4 Método 4 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando *thresholding* (MTD4)

Este é o método de segmentação adaptado de (PATTICHIS; SCHIZAS; MIDDLETON, 1995). As definições da Tabela 4 serão utilizados para descrever este método.

Uma janela deslizante de comprimento W com início em w_0 percorre o sinal da esquerda para a direita. Os BEPs dos segmentos são pontos w_0 tais que o valor máximo do sinal contido pela janela permanece abaixo do valor de *threshold* T . Para sequências de

Tabela 4: Parâmetros utilizados para definir MTD4.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal x
W	Comprimento da janela deslizante utilizada pelo método
T	Valor de <i>threshold</i>

pontos consecutivos que atendam esta especificação, seleciona-se o último ponto (ponto mais à direita). As EEPs são identificadas de forma similar, sendo os primeiros pontos $w_0 + W - 1$ após as BEPs nos quais o sinal contido pela janela permanece abaixo do valor de *threshold* T .

No trabalho de (PATTICHIS; SCHIZAS; MIDDLETON, 1995), utilizou-se janelas de comprimento W correspondente a 3 ms de aquisição e *threshold* T de $\pm 40 \mu V$ (o sinal de EMG segmentado não era retificado). O fluxograma da Figura 6 representa o MTD4.

2.3 Bases de Dados Utilizadas

Este trabalho faz uso de aquisições da base de dados número 2 da Ninapro (ATZORI et al., 2014) e a base de dados (que replica os mesmos métodos da Ninapro) em construção realizada pelo Laboratório de Instrumentação Eletro-Eletrônica (IEE).

2.3.1 Base de Dados Ninapro

2.3.1.1 Base de Dados Ninapro

2.3.2 Base de Dados do IEE

Figura 5: Fluxograma representativo do MTD3.

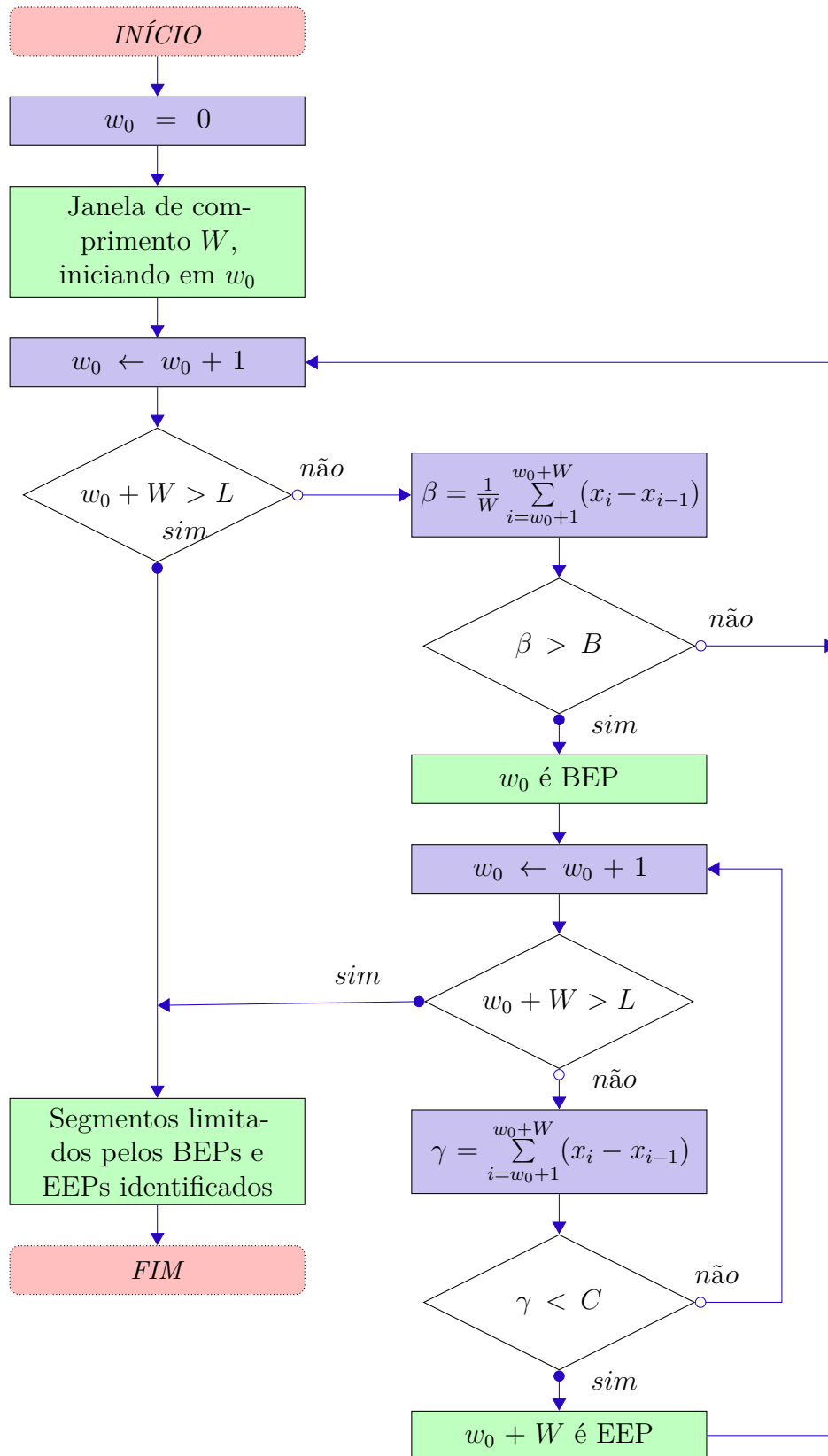
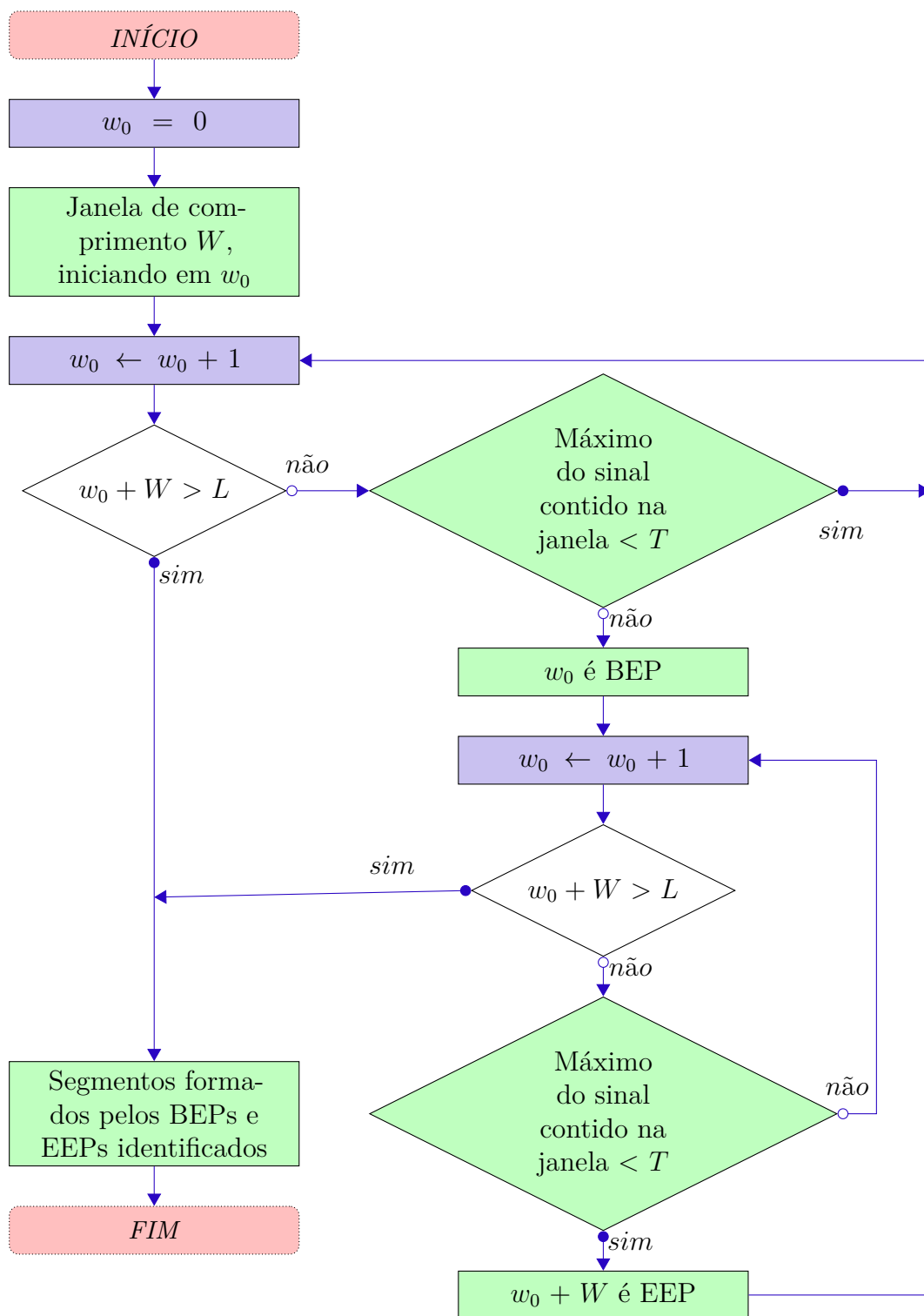


Figura 6: Fluxograma representativo do MTD4.

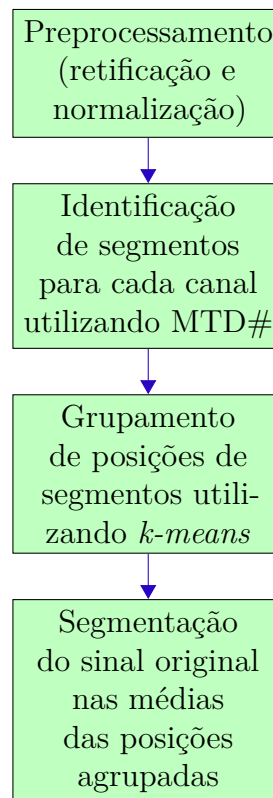


3 Metodologia Experimental

3.1 Implementação de Métodos de Segmentação

Esta seção descreve a implementação dos métodos de segmentação desenvolvidos. O fluxograma da Figura 7 apresenta de forma resumida os passos comuns aos quatro métodos, que serão explanados nas subseções seguintes. Os códigos em Matlab criados para os quatro métodos são apresentados nos Apêndices A - D.

Figura 7: Fluxograma geral para os métodos de segmentação MTD1 - MTD4.



3.1.1 Pré-processamento

3.1.1.1 Retificação Completa

Os sinais de eletromiografia para ambas as bases de dados (Ninapro e IEE) são armazenados mantendo sua polaridade original (*i.e.* amostras do sinal podem assumir valores positivos e negativos). Primeiramente, realiza-se a retificação completa dos sinais tomando o módulo dos valores amostrados (em Matlab, função `abs()`). A retificação completa do sinal mantém sua energia e é fundamental para a implementação dos métodos

de segmentação aqui desenvolvidos. A Figura 8 exemplifica o resultado esperado para a retificação completa de um trecho de sinal de eletromiografia.

Figura 8: Retificação completa de trecho de sinal de eletromiografia.



3.1.1.2 Normalização

Os sinais para cada canal de aquisição são normalizados de acordo com seu valor máximo, de modo que seu novo valor máximo seja unitário, a partir da equação 3.1, onde x é o sinal original para um canal e x_{norm} é sua versão normalizada. A normalização de canais faz com que os parâmetros utilizados pelos métodos de segmentação sejam relativos ao valor máximo do sinal, possibilitando a implementação para diferentes voluntários. A Figura 9 exemplifica a normalização para três canais de um trecho de sinal já retificado.

$$x_{norm} = \frac{x}{\max(x)} \quad (3.1)$$

3.1.2 Implementação dos Métodos de Segmentação

3.1.2.1 Parâmetros Utilizados

Cada método de segmentação MTD1 - MTD4 apresenta um conjunto de parâmetros ajustáveis. Tais parâmetros foram descritos anteriormente na seção 2.2. Após investigações iniciais das segmentações obtidas com diferentes valores de parâmetros, fixou-se alguns destes parâmetros e realizou-se uma lista de valores a serem explorados na aplicação de cada método, a fim de obter os parâmetros mais adequados para as duas bases de dados

Figura 9: Normalização de canais de eletromiografia de acordo com seu valor máximo.



utilizadas. A Tabela 5 apresenta os parâmetros ajustáveis de cada método e sua respectiva lista de valores explorados.

Tabela 5: Parâmetros ajustáveis para os métodos de segmentação.

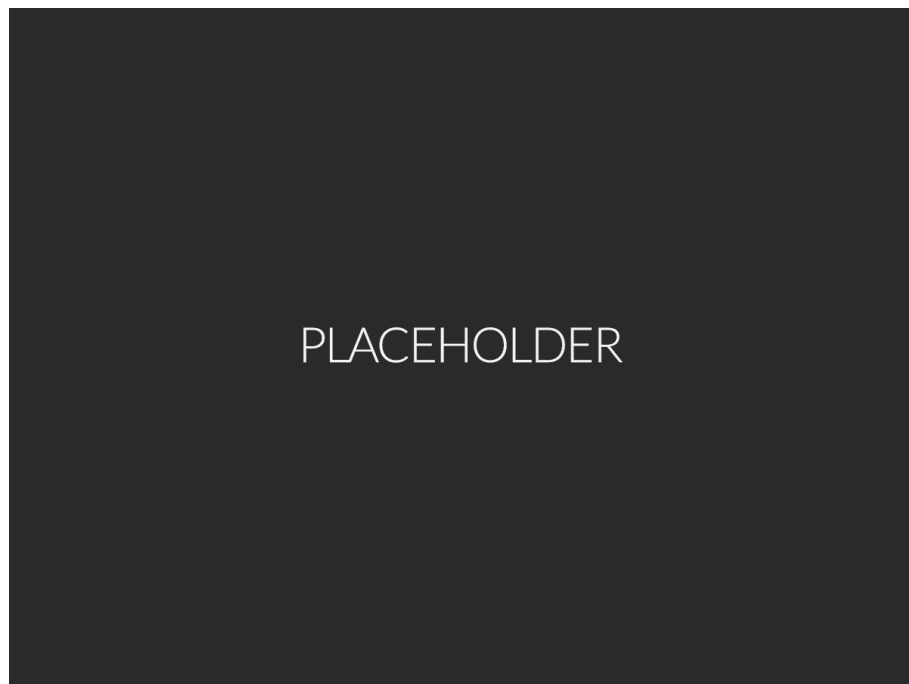
Método	Parâmetros	Valores utilizados	Número total de combinações
MTD1	l	10×10^3	16
	r_{target}	$5,6 \times 10^{-5}$	
	q	[0.8 0.85 0.9 0.95]	
	T_{lim}	[0.05 0.1 0.15 0.2]	
MTD2	l	10×10^3	27
	A	[20 30 40]	
	B	[2 5 8]	
	C	[2 5 8]	
MTD3	B	TODO	TODO
	C	TODO	
MTD4	W	TODO	TODO
	T	TODO	

Para os métodos que utilizam parâmetro de comprimento de segmento l (*i.e.* os métodos que produzem segmentos de comprimento de janela constante, MTD1 e MTD2) utilizou-se valor de l como 10×10^3 , que corresponde a 5 cinco segundos de aquisição para ambas as bases de dados (período de amostragem para ambas é de $500\mu s$), sendo a mesma duração dos segmentos de vídeo mimcados pelos voluntários. O valor de MTD1 para r_{target} foi obtido utilizando o código do Apêndice C como valor mínimo da razão entre número de segmentos e comprimento de sinal para a base de dados Ninapro.

3.1.2.2 Identificação de segmentos utilizando *k-means*

Os sinais para ambas as bases de dados são compostos por 12 canais de aquisição. Os métodos de segmentação são implementados individualmente aos doze canais. Para os métodos MTD1 e MTD2, as posições centrais dos segmentos obtidas em cada canal são armazenadas, enquanto que para os métodos MTD3 e MTD4 armazena-se as posições de BEPs e EEPs. Tais posições são agrupadas utilizando o método de *clustering k-means*. Este agrupamento permite a identificação dos segmentos obtidos nos diferentes canais que referem-se a um mesmo trecho de aumento da atividade muscular. Tomando a média de cada grupo, o sinal original pode então ser segmentado, de forma que os segmentos mantêm coerência temporal entre canais. A Figura 10 exemplifica o grupamento das posições centrais de segmentos nos métodos MTD1 e MTD2 e a Figura 12 exemplifica o grupamento de BEPs e EEPs nos métodos MTD3 e MTD4.

Figura 10: *Clustering* por *k-means* dos centros de segmentos obtidos pelos métodos MTD1 e MTD2.



3.2 Rede Neural Artificial

Esta seção descreve a utilização de RNA para classificação dos segmentos obtidos de acordo com movimentos de interesse. O processo de classificação é representado pelo fluxograma da Figura ??, que será explanado nas subseções seguintes.

Figura 11: *Clustering* por *k-means* de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.



3.2.1 Características Utilizadas como Preditores

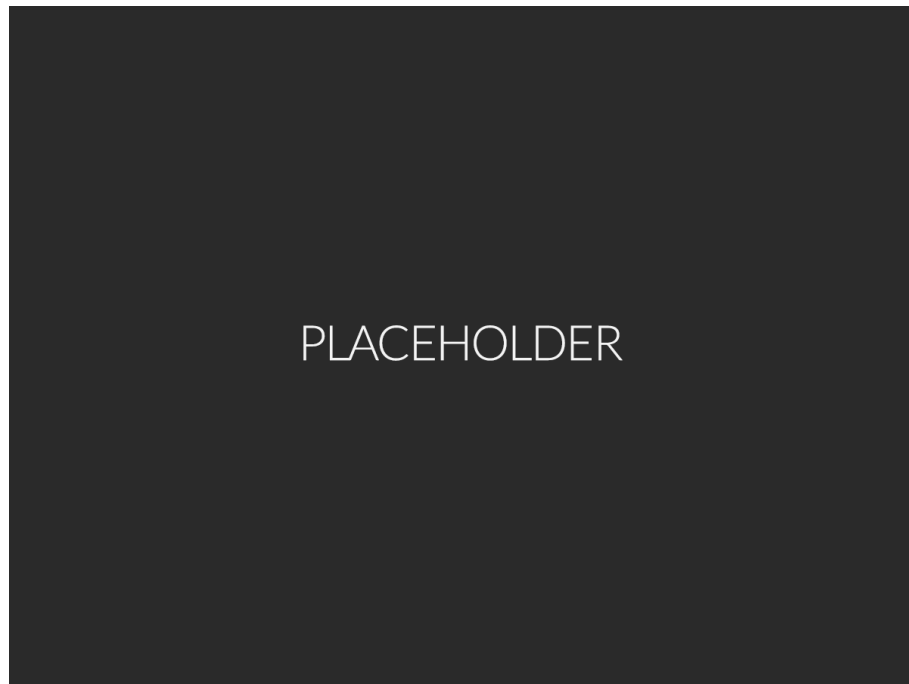
Os preditores (“entradas”) utilizados pela RNA são o valor RMS, a variância e a frequência mediana do espectro de potência dos segmentos de sinal obtidos pelos métodos MTD1 - MTD4. Tais características foram selecionados de acordo com outros trabalhos já realizados no laboratório de Instrumentação Eletro-Eletrônica que obtiveram bons resultado para uso com RNA, como (??), (??) e (??).

O valor RMS ou *root mean square* de um sinal discreto x de comprimento L (em Matlab, função $rms()$) é dada pela equação 3.2 e a variância deste sinal (em Matlab, função $var()$) é dada pela equação 3.3, onde \bar{x} é o valor médio do sinal. A frequência mediana é tal que a soma total da densidade de potência para frequências abaixo da frequência mediana é igual à soma total da densidade de potência para frequências acima da frequência mediana, sendo estimada pela função $medfreq()$ em Matlab.

$$rms(x) = \sqrt{\frac{1}{L} \sum_{i=1}^L x_i^2} \quad (3.2)$$

$$var(x) = \frac{1}{L-1} \sum_{i=1}^L (x_i - \bar{x})^2 \quad (3.3)$$

Figura 12: *Clustering* por *k-means* de BEPs e EEPs de segmentos obtidos pelos métodos MTD3 e MTD4.



3.2.2 Movimentos Utilizados como Alvo

Obtém-se os movimentos associados a cada segmento obtido a partir de sua posição temporal no sinal original, já que durante a aquisição de dados os movimentos são realizados na mesma sequência para todos os voluntários. A base de dados Ninapro apresenta um vetor *stimulus* que indica os instantes em que foram apresentados cada movimento a ser replicado pelo voluntário. A partir deste vetor e das posições dos segmentos dentro do sinal original, utiliza-se o código apresentado no Apêndice D para criar a variável que servirá como resposta esperada no treinamento da RNA.

4 Resultados e Discussões

5 Conclusões

6 Propostas de Futuros Trabalhos

Referências

- ALMEIDA, M.; BARRETO, J. *Filtragem digital de sinais biomédicos*. Universidade Federal de Santa Catarina, Centro Tecnológico., 1997. Disponível em: <<https://books.google.com.br/books?id=L5iqkgEACAAJ>>. Citado na página 15.
- ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, Macmillan Publishers Limited SN -, v. 1, p. 140053 EP -, Dec 2014. Data Descriptor. Disponível em: <<http://dx.doi.org/10.1038/sdata.2014.53>>. Citado 2 vezes nas páginas 13 e 21.
- BASMAJIAN, J.; LUCA, C. D. *Muscles Alive*. 5. ed. Baltimore: Williams and Wilkins, 1985. Citado na página 15.
- BUCHTHAL, F.; SCHMALBRUCH, H. Motor unit of mammalian muscle. *Physiological Reviews*, American Physiological Society, v. 60, n. 1, p. 90–142, 1980. Citado na página 15.
- CHAN, F. H. et al. Fuzzy EMG classification for prosthesis control. *IEEE Trans Rehabil Eng*, v. 8, n. 3, p. 305–311, Sep 2000. Citado na página 13.
- CHAUVET, E. et al. A method of emg decomposition based on fuzzy logic. In: *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*. [S.l.: s.n.], 2001. v. 2, p. 1948–1950 vol.2. ISSN 1094-687X. Citado 2 vezes nas páginas 13 e 17.
- CHU, J.-U. et al. A supervised feature-projection-based real-time emg pattern recognition for multifunction myoelectric hand control. *Mechatronics, IEEE/ASME Transactions on*, v. 12, n. 3, p. 282–290, June 2007. ISSN 1083-4435. Citado na página 13.
- ENGLEHART, K.; HUDGINS, B. A robust, real-time control scheme for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, v. 50, n. 7, p. 848–854, July 2003. ISSN 0018-9294. Citado na página 13.
- GERDLE, B. et al. Acquisition, processing and analysis of the surface electromyogram. In: WINDHORST, U.; JOHANSSON, H. (Ed.). *Modern Techniques in Neuroscience Research*. Springer Berlin Heidelberg, 1999. p. 705–755. ISBN 978-3-642-63643-1. Disponível em: <http://dx.doi.org/10.1007/978-3-642-58552-4_26>. Citado na página 16.
- GIJSBERTS, A. et al. Measuring movement classification performance with the movement error rate. *IEEE Transactions on neural systems and rehabilitation engineering*, 2014. Citado na página 14.
- GUT, R.; MOSCHYTZ, G. S. High-precision emg signal decomposition using communication techniques. *Signal Processing, IEEE Transactions on*, v. 48, n. 9, p. 2487–2494, Sep 2000. ISSN 1053-587X. Citado 2 vezes nas páginas 14 e 20.
- HARGROVE, L. J. et al. Robotic leg control with emg decoding in an amputee with nerve transfers. *New England Journal of Medicine*, v. 369, n. 13, p. 1237–1242, 2013. PMID: 24066744. Disponível em: <<http://dx.doi.org/10.1056/NEJMoa1300126>>. Citado na página 13.

HUDGINS, B.; PARKER, P.; SCOTT, R. A new strategy for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, v. 40, n. 1, p. 82–94, Jan 1993. ISSN 0018-9294. Citado na página 13.

KATSIS, C. et al. A novel method for automated {EMG} decomposition and {MUAP} classification. *Artificial Intelligence in Medicine*, v. 37, n. 1, p. 55 – 64, 2006. ISSN 0933-3657. Intelligent Data Analysis in Medicine. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0933365705001065>>. Citado 3 vezes nas páginas 14, 18 e 19.

KLINE, J. C.; LUCA, C. J. D. Error reduction in emg signal decomposition. *Journal of Neurophysiology*, American Physiological Society, 2014. ISSN 0022-3077. Citado na página 16.

LUCA, C. J. D. et al. Decomposition of surface EMG signals. *J. Neurophysiol.*, v. 96, n. 3, p. 1646–1657, Sep 2006. Citado na página 15.

PATTICHIS, C.; SCHIZAS, C.; MIDDLETON, L. Neural network models in emg diagnosis. *Biomedical Engineering, IEEE Transactions on*, v. 42, n. 5, p. 486–496, May 1995. ISSN 0018-9294. Citado 5 vezes nas páginas 13, 14, 16, 20 e 21.

Apêndices

APÊNDICE A – Função em Matlab para MTD1

```
function [x_seg, finalCenterLocs] = seg_mtd1(x, l, q, r_target, T_lim)
% MTD1 - metodo iterativo utilizando thresholding para deteccao de
% centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD1)
% x - matriz cujas colunas sao canais do sinal a ser segmentado
% l - comprimento desejado para os segmentos
%      (deve ser inteiro maior que zero)
% q - razao de atualizacao entre iteracoes para valor de threshold
%      (deve ser entre 0 e 1)
% r_target - razao minima esperada entre numero de segmentos e comprimento
%      total de sinal
%      (deve ser maior que zero)
% T_lim - fracao do maximo do sinal para limite inferior de threshold
%      (deve ser entre 0 e 1)
%
% Retorno:
% x_seg - cell array com canais segmentados
% finalCenterLocs - posicoes centrais dos segmentos

%% Preprocessamento

[L, numberOfChannels] = size(x); % comprimento do sinal e numero de canais
x_ret = abs(x); % retificacao
x_norm = zeros(L, numberOfChannels); % normalizacao
for currentChannel = 1:numberOfChannels
    x_norm(:,currentChannel) = ...
        x_ret(:,currentChannel)./(max(x_ret(:,currentChannel)));
end

%% Metodo

centerLocsCell = cell(1,numberOfChannels);
for currentChannel = 1:numberOfChannels
    T_k = 1; % canais normalizados, seus valores maximos sempre sao 1
    targetReached = false;
    while ~targetReached % processo iterativo
        T_k = q*T_k; % calcula threshold desta iteracao
        if T_k < T_lim % verifica se o limite de valor de threshold foi atingido
```

```

        break
    end
    % Identifica candidatos a centros de segmentos
    [~, centerLocsCell{1,currentChannel}] = ...
        findpeaks(double(x_norm(:,currentChannel)), ...
        'MinPeakHeight', T_k, 'MinPeakDistance',1);
    % Determina o encerramento do processo iterativo
    targetReached = (length(centerLocsCell{1,currentChannel})/L >= r_target);
end
end

%% Clustering

centerLocsArray = sort(cell2mat(centerLocsCell'));
[~, labscore] = dbscan(centerLocsArray,2000,3);
numberOfSegments = max(labscore);
finalCenterLocs = zeros(numberOfSegments,1); % medias internas aos clusters
for currentCluster = 1:numberOfSegments
    finalCenterLocs(currentCluster) = ...
        round(mean(centerLocsArray(labscore == currentCluster)));
end

%% Segmentacao

x_seg = cell(numberOfSegments, numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        switch mod(1,2)
            case 0 % se l for par
                if(finalCenterLocs(currentSegment)-1/2)<1
                    % segmento muito a esquerda
                    x_seg{currentSegment,currentChannel} = ...
                        x(1:finalCenterLocs(currentSegment)+(1/2)-1, ...
                        currentChannel);
                else if(finalCenterLocs(currentSegment)+(1/2)-1)>L
                    % segmento muito a direita
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2:L, ...
                        currentChannel);
                else
                    x_seg{currentSegment,currentChannel} = ...
                        x(finalCenterLocs(currentSegment)-1/2: ...
                        finalCenterLocs(currentSegment)+(1/2)-1, ...
                        currentChannel);
                end
            end
            case 1 % se l for impar

```

```
if(finalCenterLocs(currentSegment) - (l-1)/2)<1
    % segmento muito a esquerda
    x_seg{currentSegment,currentChannel} = ...
        x(1:finalCenterLocs(currentSegment) + (l-1)/2, ...
            currentChannel);
else if(finalCenterLocs(currentSegment) + (l-1)/2)>L
    % segmento muito a direita
    x_seg{currentSegment,currentChannel} = ...
        x(finalCenterLocs(currentSegment) - (l-1)/2:L, ...
            currentChannel);
else
    x_seg{currentSegment,currentChannel} = ...
        x(finalCenterLocs(currentSegment) - (l-1)/2: ...
            finalCenterLocs(currentSegment) + (l-1)/2, ...
            currentChannel);
end
end
end
end
end
end
```

APÊNDICE B – Função em Matlab para MTD2

```

%%
%   MTD2 - metodo nao iterativo utilizando thresholding para deteccao de
%   centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD2)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   l - comprimento desejado para os segmentos
%       (deve ser inteiro maior que zero)
%   A - coeficiente utilizado para decisao de metodo de calculo de threshold
%       (deve ser maior ou igual a 1)
%   B - multiplo da media aritmetica do sinal x para obtencao de threshold
%       (deve ser maior ou igual a 1)
%   C - fracao do valor maximo do sinal x para calculo de threshold
%       (deve ser maior ou igual a 1)
%
% Retorno:
%   x_seg - cell array com os canais segmentados
%   centerLocs - posicoes centrais dos segmentos
%%

function [x_seg, centerLocs] = seg_mtd2(x, l, A, B, C)

%% Preprocessamento

% Obtem comprimento do sinal e numero de canais
[L, numberOfChannels] = size(x);

% Retificacao
x_ret = abs(x);

% Normalizacao
x_norm = zeros(L, numberOfChannels);
for currentChannel = 1:numberOfChannels
    x_norm(:,currentChannel) = ...
        x_ret(:,currentChannel) ./ max(x_ret(:,currentChannel));
end

%% Metodo

```



```

% Cell array para armazenar posicoes dos segmentos identificados
centerLocsCell = cell(1,numberOfChannels);

for currentChannel = 1:numberOfChannels

    % Calculo do threshold
    maxValue = 1; % sinais normalizados
    meanValue = mean(x_norm(:,currentChannel));
    if maxValue > (A*meanValue)
        T = B*meanValue;
    else
        T = maxValue/C;
    end

    % Identifica centros de segmentos
    [~, centerLocsCell{1,currentChannel}] = ...
        findpeaks(double(x_norm(:,currentChannel)), ...
            'MinPeakHeight', T, 'MinPeakDistance',1);
end

%% Segmentacao dos canais

% Maximo numero de segmentos detectados
numberOfSegments = 0;
for currentChannel = 1:numberOfChannels
    currentChannelNumberOfSegments = length(centerLocsCell{1,currentChannel});
    if currentChannelNumberOfSegments > numberOfSegments
        numberOfSegments = currentChannelNumberOfSegments;
    end
end

% Clustering dos centros de segmentos detectados
centerLocsArray = cell2mat(centerLocsCell');
idx = kmeans(centerLocsArray,numberOfSegments);

% Clustering dos centros de segmentos detectados
centerLocsArray = cell2mat(centerLocsCell');
[~,C] = kmeans(centerLocsArray,numberOfSegments);
centerLocs = sort(round(C));

% Segmentacao
x_seg = cell(numberOfSegments,numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        if mod(1,2) == 0 % se l for par
            if (centerLocs(currentSegment)-1/2) < 1 % segmento muito a esquerda
                x_seg{currentSegment,currentChannel} = ...

```

```

        x(1:centerLocs(currentSegment)+l/2 - 1, currentChannel);
    else if (centerLocs(currentSegment)+l/2 - 1) > L % segmento muito a direita
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2:L, currentChannel);
    else
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2: ...
                centerLocs(currentSegment)+l/2 - 1, currentChannel);
    end
end
else % se l for impar
    if (centerLocs(currentSegment)-(l+1)/2) < 1 % segmento muito a esquerda
        x_seg{currentSegment,currentChannel} = ...
            x(1:centerLocs(currentSegment)+l/2 - 1, currentChannel);
    else if (centerLocs(currentSegment)+(l+1)/2 - 1) > L % segmento muito a direita
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2:L, currentChannel);
    else
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-(l+1)/2: ...
                centerLocs(currentSegment)+(l+1)/2 - 1, currentChannel);
    end
end
end
end
end
end
end

```

APÊNDICE C – Código utilizado para determinação de r_{target} do MTD1

```
%%
%   Determinacao de r_target para MTD1
%%

%% Base de dados Ninapro

ninaproList = ls('database/ninapro2/S*_E1*');
numberOfSubjects = length(ninaproList);
numberOfSegments = 102; % 17 movimentos * 6 repeticoes

%% Obtencao de r_target da database

r_target = zeros(numberOfSubjects, 1);

for currentSubject = 1:numberOfSubjects

    fprintf('currentSubject = %i\n',currentSubject)

    % Carrega o voluntario atual
    load (['database/ninapro2/' ninaproList(currentSubject,:)])

    % r_target para este voluntario
    r_target(currentSubject) = numberOfSegments/length(emg);

end

min_r_target = min(r_target)
```

APÊNDICE D – Função em Matlab para obtenção dos movimentos relacionados aos segmentos

```

%%
%  Identificacao das classes de movimentos
%%

function targetDataArray = identifyClasses(centerLocs, stimulus)

%% Remodela o vetor de estimulo, removendo trechos de 0

reshapedStimulus = stimulus;
currentMovement = 1;
for index = 1:length(stimulus)
    if reshapedStimulus(index) ~= 0
        currentMovement = reshapedStimulus(index);
    else
        reshapedStimulus(index) = currentMovement;
    end
end

%% Gera a matriz de identificacao das classes

numberOfSegments = length(centerLocs);
numberOfClasses = 17;
targetDataArray = false(numberOfSegments, numberOfClasses);
for currentSegment = 1:numberOfSegments
    targetDataArray(currentSegment,...
        reshapedStimulus(centerLocs(currentSegment))) = true;
end

end

```